**W200 PROJECT 1 REFLECTION - SLAP**

INSTRUCTIONS: You start the game by running W200proj1.py in the terminal. For this program you do not hit enter after typing something. There are instructions for the rules of the game that pop up first, but the main points are: press ‘d’ to add a card, press ‘s’ to slap, watch out for Doubles or Sandwiches. If you press ‘s’ when it is not a slap, a message will appear telling you you lost three cards. If you press capital ‘D’ or any key without a ‘function’ attached to it, you’ll get warning messages that will stay on screen and multiply if you continue pressing the wrong keys. I realized today that if you have pressed the wrong key too many times, an error happens I think because the error messages take up more space than the window allows. I tried to fix this, but right now it still gets an error on my computer. Every time cards are added to the discard pile, all the decks and deck lengths are updated on screen.  After a slap event, a message will appear telling you who won and whether it was a double or a sandwich. However, if it is the computer’s turn next, the message won’t stay on the screen for long, so you need to keep that in mind. If it is your turn next, the game is waiting for you to add a card, so the message stays up until you do so. The Window object closes if you quit or the game ends.

I started this project with a good idea of what I wanted/ what to code. The card and initial deck classes were easy to build. When I started coding the user input, I hit my first snag. I realized that I couldn’t use the input() function for my program because you had to press ‘Enter’, which meant that the game could not be fast and reactive like I had hoped. Since the instructions said that we needed to stick to the python standard library, I tried to find a package in it that would allow for quick time events like I wanted. I found the curses library. I hadn’t used it before and spent about a day trying to figure it out. I actually thought at one point that I would have to scrap my entire project design because I was having so much trouble. I reached my second snag when I realized that using curses and Window objects meant that I could not use the print() function and instead use the curses functions for changing the window’s display. It also took me a while to realize the importance of using the curses.refresh() function every time I wanted the window display to change. I ended up creating my classes and functions when I wanted to reduce the amount of repeated code. I’m glad I did this because it was much easier to try different window displays and pause times after that. I initially added the number of cards in the player’s decks to the screen as a way to test the game and debug, but I realized that for double events, having the number of cards displayed made helped the player recognize Double events, so I kept it in. For some reason that I never figured out, I had a really hard time programming the sandwich event. I actually removed it at one point, but realized I needed more lines of code, and decided to retry. Then miraculously it worked, and I have no idea what changed.

I do have ideas of what I might add. I probably would have added some of them if I had had say another day, but I was more focused on hitting the requirements. Gerry asked me if I had considered adding a ‘beep’ sound to play during the double event. I had found that option when reading through curse and window python documentation, but I had decided against it thinking that the visual was easier to identify. Now I wonder if I could have used a combination of the two or used the beep to make the game more accessible. I also probably would have played around more with the text display and see how to increase the size and make it easier to read for people with strained eyesight like my parents (who tested the game for me once before I remembered that they need to hold their phones right up to their faces to read it).